LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-206841

3-D Combinatorial Geometry in the MERCURY Monte Carlo Particle Transport Code

G. Greenman

September 27, 2004

Nuclear Explosives Code Developers Conference 2004
Livermore, CA, United States
October 4, 2004 through October 7, 2004

3-D Combinatorial Geometry in the MERCURY Monte Carlo Particle Transport Code (U)

Gregory Greenman

Lawrence Livermore National Laboratory
Mail Stop L-95, P.O. Box 808
Livermore, CA 94551

Lawrence Livermore National Laboratory is in the process of developing a new Monte Carlo Particle Transport code named MERCURY. This new code features a 3-D Combinatorial Geometry tracking algorithm. This paper details some of the characteristics of this Monte Carlo tracker

Introduction

The Lawrence Livermore National Laboratory has been using codes based on the Monte Carlo Transport Method toto do particle transport calculations in support of Laboratory programs. MERCURY is the latest development in the stable of transport calculational capabilities at the Laboratory.

Many programs executed at the Lawrence Livermore National Laboratory require the calculation of the behavior and transport of neutrons, photons, and charged particles in various media and their interaction with the same. To this end, over the years the Laboratory has supported the development of a number of transport codes including those based on the Monte Carlo Method. TART and COG are examples of two codes based on the Monte Carlo Method.

Although TART and COG have served the Laboratory well over the many years of their tenure, it is timely that the Laboratory update its capabilities in the field. TART and COG were written in FORTRAN – which was the standard language for scientific codes. However, the more modern codes are now written in the “C” programing language. “C” offers many advantages including greater “portability” - which is premium when computing platforms and architectures are changed as rapidly as is done currently. Additionally, the “C” programming standard includes a number of very useful concepts such as “pointered” arrays. This concept is intrinsic to “C”, but had to be “kludged” into the various FORTRAN compilers, since it was not originally in the language definition.

Since the middle 1990s, and the start of the Accelerated Strategic Computing Initiative [ASCI], the Laboratory has been utilizing the latest advances in computing hardware – name Massively Parallel Architectures. Older codes, like TART and COG were conceived before the advent of these Massively Parallel Architectures. They have had to be retro-fitted in order to take maximal advantage of these new architectures. Conversely, MERCURY was conceived specifically with these Massively Parallel Architectures in mind, and therefore the ability to make maximal use of parallel processing was engineered into MERCURY from the beginning.

Basic Properties of 3-D Combinatorial Geometry

Originally, MERCURY modeled its computational geometry with mesh. In many applications, especially those in which material properties show variation on very short length scales, a mesh representation is the most efficient way of representing the geometry and its materials. However, when the geometry has large homogeneous regions, a mesh-based model is not the most efficient way to represent the geometry. In these cases, a Combinatorial Geometry is much more efficient. In Combinatorial Geometry, the object under analysis is modeled as a collection of "cells" - those cells being defined by their bounding surfaces. These bounding surfaces are represented via the equation of "analytic geometry" that defines the locus of all points on that surface.

In addition to primitive cells which are defined by their bounding surfaces, MERCURY will also allow the specification of a cell by a logical aggregation using the logical operators, "AND", "OR", and "NOT" applied to collections of primitive cells.

It is also intended that MERCURY will be able to mix both mesh-based and Combinatorial Geometry based geometric representations of the objects under analysis, in order to take advantage of the benefits of each technique where they are appropriate.

Surfaces and Cells

In terms of the types of surfaces that MERCURY can model, the code most certainly is able to represent "x-normal", "y-normal", and "z-normal" planes, as well as planes of arbitrary orientation specified by either user-input points on the planes, or user-input coefficients. Arbitrarily oriented planes may also be realized by starting with an axis normal oriented plane, and rotating it via one of the rotational operators that MERCURY supports.

MERCURY also supports a suite of 2^{nd} order surfaces. These include, spheres, ellipsoids, hyperboloids, elliptic paraboloids, and other surfaces that can be described by a 2^{nd} order equation.

Additionally, MERCURY supports cones and cylinders, as well as a variety of multi-faceted surfaces such as tetrahedrons, pentahedrons, hexahedrons, pyramids, and prisms. It is also intended that MERCURY support the solid modeling surface specification known as NURBS.

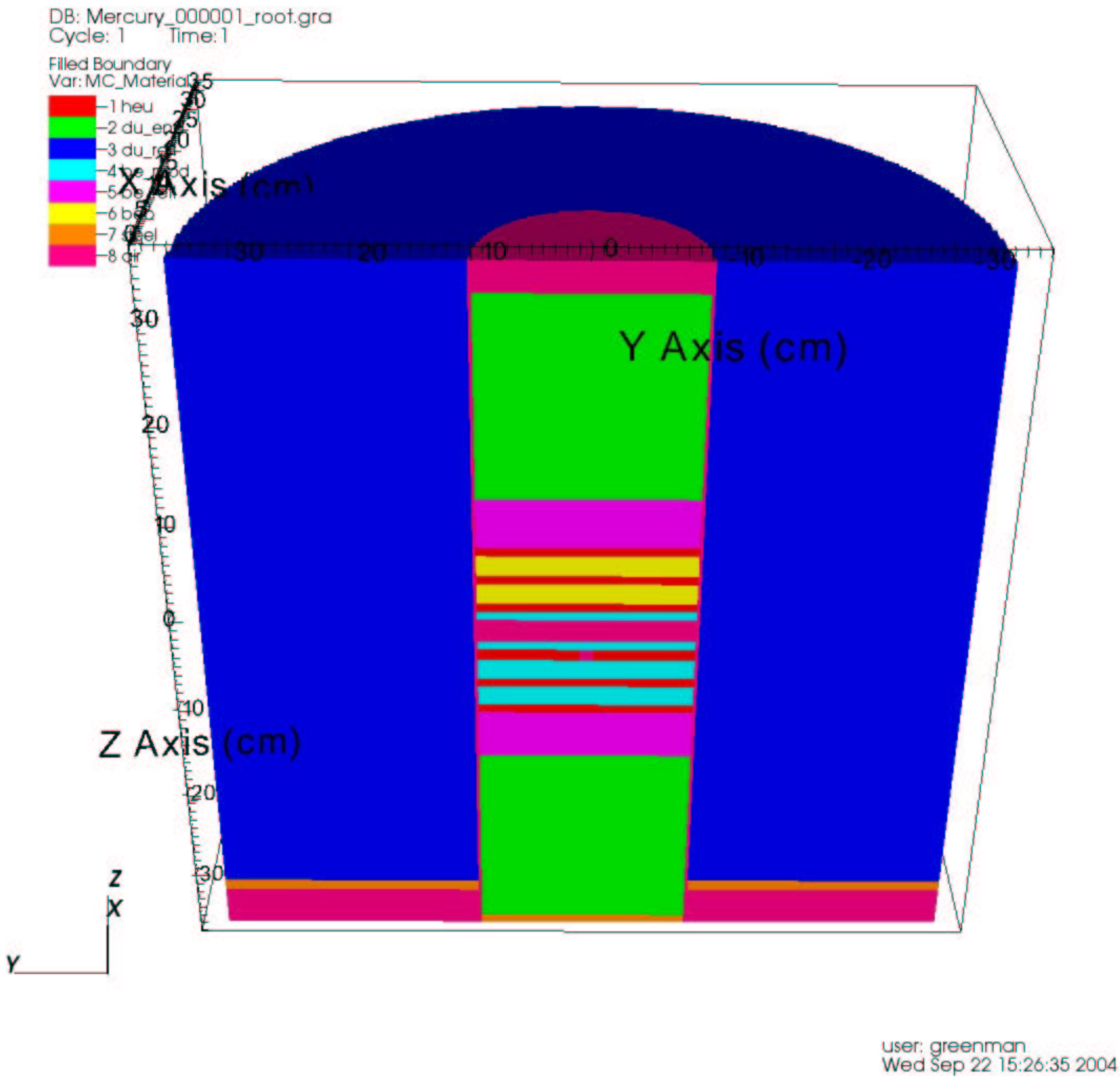
In order to specify a well-posed problem, the user only has to define the surfaces, and how these surfaces bound the “cells”, the volumes that are of interest. Unlike some codes, MERCURY relieves the user from the task of specifying the “connectivity” - that is “what cells bound what other cells”. MERCURY does this automatically during the course of tracking particles, and builds its own “connectivity tree”, “on the fly”.

The ability to build these “connectivity trees” relieves the code of much of the burden of searching for neighboring cells. Candidate cells in the tree can be searched first. If none of the candidate cells encompasses the particle's new location, because no particle has visited this cell before, a more exhaustive search is made of the entire universe of cells. Various techniques are employed to make these searches as efficiently as possible in order to maximize overall computational efficiency.

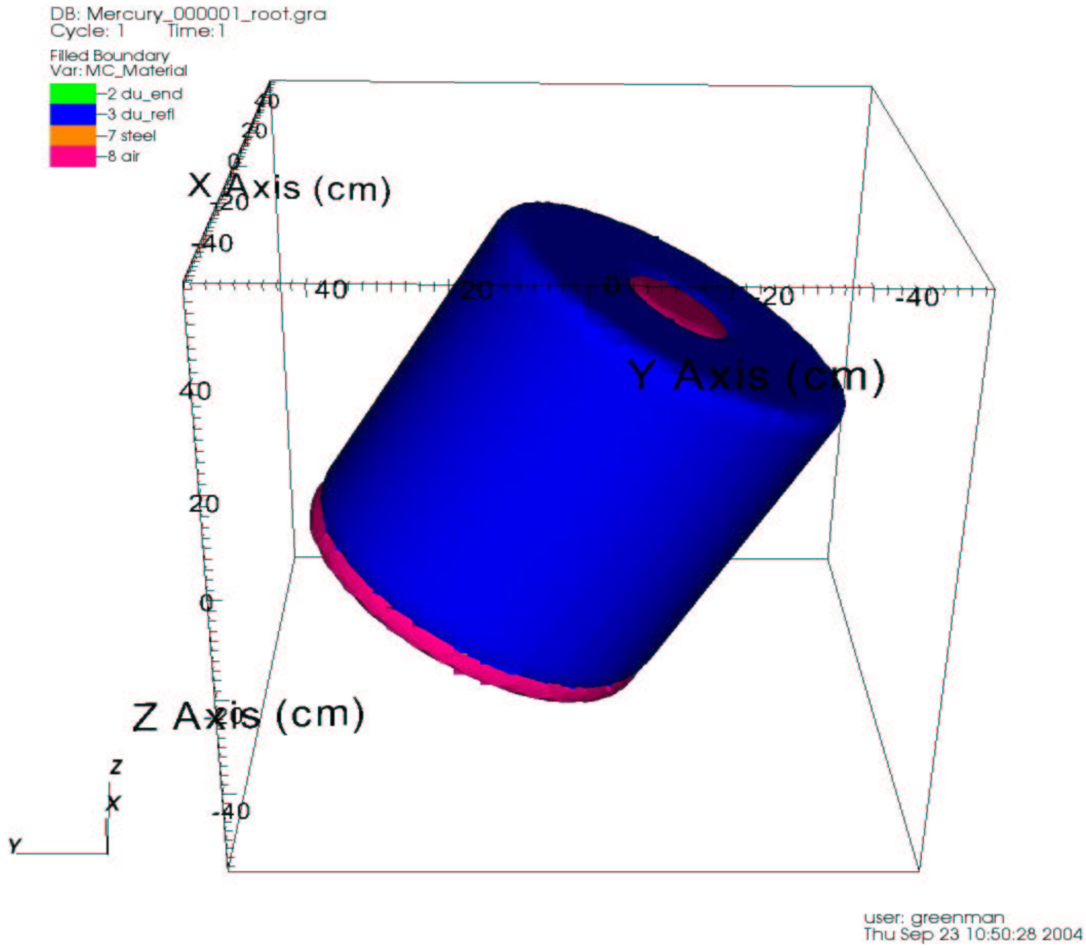
Verification and Validation - “*V&V*”

As part of the “Verification and Validation” [“V&V”] regime required of the new modern codes, MERCURY is being extensively tested. One such test problem is to be found in the *International Handbook of Evaluated Criticality Safety Benchmark Experiments* which is a collection of the specifications of many systems that have been experimentally verified to be exactly critical.

One such problem is the “HEU-38” benchmark. This benchmark problem consists of a number of discs of Highly Enriched Uranium [HEU] interspersed in a stack with other discs containing moderating materials; either Beryllium or Beryllium Oxide. The stack is reflected axially by cylinders of Depleted Uranium [DU], and surrounded by cylinder of DU to provide radial reflection. The entire assembly rests on a disc of stainless steel. Note also that the tiny air gaps that exist between the various parts is explicitly modeled in the Combinatorial Geometry representation.


DB: Mercury_000001_root.gra
Cycle: 1 Time:1
Filled Boundary
Var: MC_Material
1 heu
X Axis (cm)
Y Axis (cm)
Z Axis (cm)
user: greenman
Wed Sep 22 15:26:35 2004

As alluded to previously, MERCURY allows to user to perform a variety of rotations and translations on any defined object, in order to make the definition of the problem at hand as easy as possible.

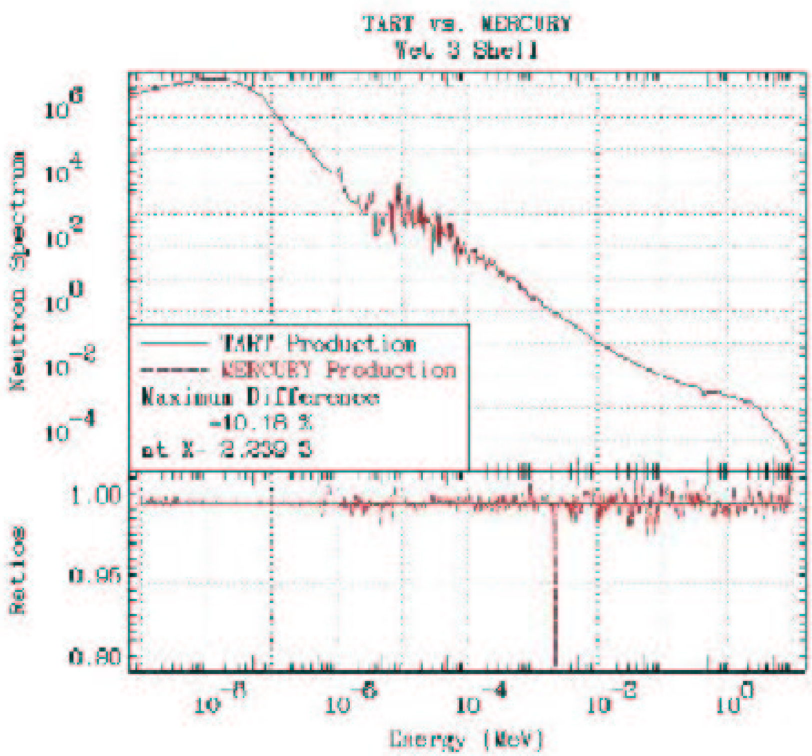


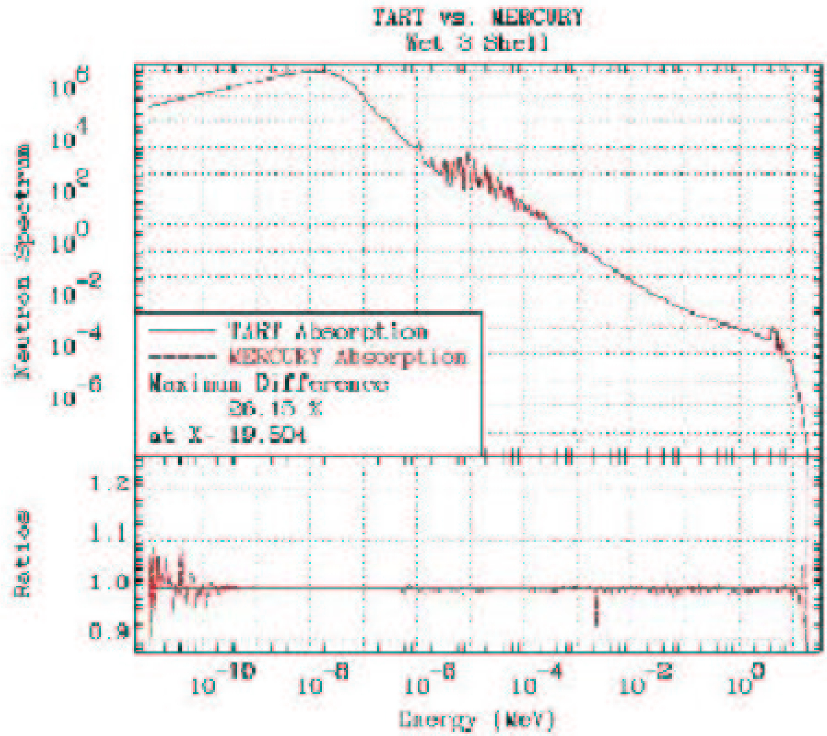
In pursuit of the Validation and Verification effort, MERCURY results are being compared to those produced by the TART code, whose results are well accepted.

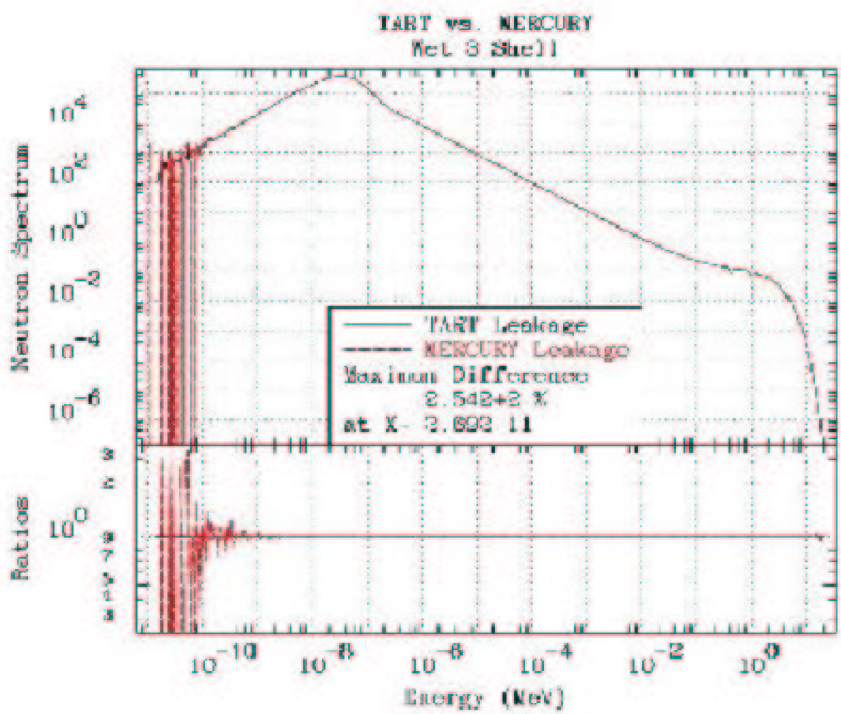
In addition the HEU-38 benchmark problem above, MERCURY and TART results are also being compared for the Godiva and Jezebel bare sphere critical assemblies. An additional testing platform are the so-called “Broomstick” problems. These problems consist of extremely long – but extremely thin cylinders. Any “anomalous” behavior by a code in tracking neutrons through said geometry can result in a neutron departing the problem when it shouldn't. Therefore, the code doesn't get a chance to correct of “anneal” its errors.

In order to validate the neutron thermalization models, a variant, each, of the Godiva, Jezebel, and “3 Shell” problems where defined in which the material each assembly is made of is defined as including 100 parts of water to each part heavy metal. The water serves to thermalize the neutron spectrum. These are called the “Wet Godiva”, “Wet Jezebel” , and “Wet 3 Shell” problems, respectively.

I




TART vs. MERCURY
Wet 3 Shell
TART Absorption
MERCURY Absorption
Maximum Difference
26.15 %
at X- 19.504
Neutron Spectrum
Ratios
Energy (MeV)


TART vs. MERCURY
Wet 3 Shell
TART Leakage
MERCURY Leakage
Maximum Difference
2.542+2 %
Neutron Spectrum
Ratios
Energy (MeV)

Summary

In summary, the new, modern Monte Carlo code MERCURY has a working Combinatorial Geometry tracker. This capability is still being extended to provide enhanced transport capabilities. Verification and Validation of the code are proceeding apace, and will shortly be able to support the ongoing programs at Lawrence Livermore that require particle transport capabilities.

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.